PATENT APPLICATION
ATTORNEY DOCKET NO. 13220/010001; P5845

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: **REPLICATION ARCHITECTURE FOR A DIRECTORY SERVER**

APPLICANT: **John MERRELLS, Olga NATKOVICH, Gordon GOOD, Rich MEGGINSON, Ludovic POITOU and Mark C. SMITH**

REPLICATION ARCHITECTURE FOR A DIRECTORY SERVER

Background of Invention

[0001] The most fundamental program resident on any computer is the operating system (OS). Various operating systems exist in the market place, including Solaris™ from Sun Microsystems Inc., Palo Alto, CA (Sun Microsystems), MacOS from Apple Computer, Inc., Cupertino, CA, Windows® 95/98 and Windows NT®, from Microsoft Corporation, Redmond, WA, UNIX, and Linux. The combination of an OS and its underlying hardware is referred to herein as a "traditional platform". Prior to the popularity of the Internet, software developers wrote programs specifically designed for individual traditional platforms with a single set of system calls and, later, application program interfaces (APIs). Thus, a program written for one platform could not be run on another. However, the advent of the Internet made cross-platform compatibility a necessity and a broader definition of a platform has emerged. Today, the original definition of a traditional platform (OS/hardware) dwells at the lower layers of what is commonly termed a "stack," referring to the successive layers of software required to operate in the environment presented by the Internet and World Wide Web.

[0002] Effective programming at the application level requires the platform concept to be extended all the way up the stack, including all the new elements introduced by the Internet. Such an extension allows application programmers to operate in a stable, consistent environment.

[0003] iPlanet™ E-commerce Solutions, a Sun Microsystems|Netscape Alliance, has developed a net-enabling platform shown in Figure 1 called the Internet Service Deployment Platform (ISDP) (28). ISDP (28) gives businesses a very

broad, evolving, and standards-based foundation upon which to build an e-enabled solution.

[0004] A core component of the ISDP (28) is iPlanet™ Directory Server (80), a Lightweight Directory Access Protocol (LDAP)-based solution that can handle more than 5,000 queries per second. iPlanet™ Directory Server (iDS) provides a centralized directory service for an intranet or extranet while integrating with existing systems. The term "directory service" refers to a collection of software, hardware, and processes that store information and make the information available to users. The directory service generally includes at least one instance of the iDS and one or more directory client program(s). Client programs can access names, phone numbers, addresses, and other data stored in the directory.

[0005] The iDS is a general-purpose directory that stores all information in a single, network-accessible repository. The iDS provides a standard protocol and application programming interface (API) to access the information contained by the iDS. The iDS provides global directory services, meaning that information is provided to a wide variety of applications. Until recently, many applications came bundled with a proprietary database. While a proprietary database can be convenient if only one application is used, multiple databases become an administrative burden if the databases manage the same information. For example, in a network that supports three different proprietary e-mail systems where each system has a proprietary directory service, if a user changes passwords in one directory, the changes are not automatically replicated in the other directories. Managing multiple instances of the same information results in increased hardware and personnel costs.

[0006] The global directory service provides a single, centralized repository of directory information that any application can access. However, giving a wide variety of applications access to the directory requires a network-based means of

communicating between the numerous applications and the single directory. The iDS uses LDAP to give applications access to the global directory service.

[0007] LDAP is the Internet standard for directory lookups, just as the Simple Mail Transfer Protocol (SMTP) is the Internet standard for delivering e-mail and the Hypertext Transfer Protocol (HTTP) is the Internet standard for delivering documents. Technically, LDAP is defined as an on-the-wire bit protocol (similar to HTTP) that runs over Transmission Control Protocol/Internet Protocol (TCP/IP). LDAP creates a standard way for applications to request and manage directory information.

[0008] An LDAP-compliant directory, such as the iDS, leverages a single, master directory that owns all user, group, and access control information. The directory is hierarchical, not relational, and is optimized for reading, reliability, and scalability. This directory becomes the specialized, central repository that contains information about objects and provides user, group, and access control information to all applications on the network. For example, the directory can be used to provide information technology managers with a list of all the hardware and software assets in a widely spanning enterprise. Most importantly, a directory server provides resources that all applications can use, and aids in the integration of these applications that have previously functioned as stand-alone systems. Instead of creating an account for each user in each system the user needs to access, a single directory entry is created for the user in the LDAP directory. Figure 2 shows a portion of a typical directory with different entries corresponding to real-world objects. The directory depicts an organization entry (90) with the attribute type of domain component (dc), an organizational unit entry (92) with the attribute type of organizational unit (ou), a server application entry (94) with the attribute type of common name (cn), and a person entry (96) with the attribute type of user ID (uid). All entries are connected by the directory.

[0009] Understanding how LDAP works starts with a discussion of an LDAP protocol. The LDAP protocol is a message-oriented protocol. The client constructs an LDAP message containing a request and sends the message to the server. The server processes the request and sends a result, or results, back to the client as a series of LDAP messages. Referring to Figure 3, when an LDAP client (100) searches the directory for a specific entry, the client (100) constructs an LDAP search request message and sends the message to the LDAP server (102) (step 104). The LDAP server (102) retrieves the entry from the database and sends the entry to the client (100) in an LDAP message (step 106). A result code is also returned to the client (100) in a separate LDAP message (step 108).

[0010] LDAP-compliant directory servers like the iDS have nine basic protocol operations, which can be divided into three categories. The first category is interrogation operations, which include search and compare operators. These interrogation operations allow questions to be asked of the directory. The LDAP search operation is used to search the directory for entries and retrieve individual directory entries. No separate LDAP read operation exists. The second category is update operations, which include add, delete, modify, and modify distinguished name (DN), *i.e.*, rename, operators. A DN is a unique, unambiguous name of an entry in LDAP. These update operations allow the update of information in the directory. The third category is authentication and control operations, which include bind, unbind, and abandon operators.

[0011] The bind operator allows a client to identify itself to the directory by providing an identity and authentication credentials. The DN and a set of credentials are sent by the client to the directory. The server checks whether the credentials are correct for the given DN and, if the credentials are correct, notes that the client is authenticated as long as the connection remains open or until the client re-authenticates. The unbind operation allows a client to terminate a session. When the client issues an unbind operation, the server discards any

authentication information associated with the client connection, terminates any outstanding LDAP operations, and disconnects from the client, thus closing the TCP connection. The abandon operation allows a client to indicate that the result of an operation previously submitted is no longer of interest. Upon receiving an abandon request, the server terminates processing of the operation that corresponds to the message ID.

[0012] In addition to the three main groups of operations, the LDAP protocol defines a framework for adding new operations to the protocol via LDAP extended operations. Extended operations allow the protocol to be extended in an orderly manner to meet new marketplace needs as they emerge.

[0013] The basic unit of information in the LDAP directory is an entry, a collection of information about an object. Entries are composed of a set of attributes, each of which describes one particular trait of an object. Attributes are composed of an attribute type (*e.g.*, common name (cn), surname (sn), etc.) and one or more values. Figure 4 shows an exemplary entry (124) showing attribute types (120) and values (122). Attributes may have constraints that limit the type and length of data placed in attribute values (122). A directory schema places restrictions on the attribute types (120) that must be, or are allowed to be, contained in the entry (124).

Summary of Invention

[0014] In general, in one aspect, the invention involves a directory server. The directory server comprises a supplier server, a consumer server in communication with the supplier server, a plurality of pluggable services that manage replication of data contained within the directory server from the supplier server to the consumer server, and a change log maintained on the consumer server of data

replicated to the consumer server. Replication of data is managed by the plurality of pluggable services using the change log.

[0015] In general, in one aspect, the invention involves a method for replicating data in a directory server having a supplier and a consumer server. The method comprises determining a need to replicate data in the directory server, using a plurality of services to manage replication of data contained within the directory server from the supplier server to the consumer server, maintaining a change log of data replicated to the consumer server, and updating data replicated to the consumer server.

[0016] In general, in one aspect, the invention involves a method for replicating data in a directory server having a supplier and a consumer server. The method comprises determining a need to replicate data in the directory server, using a plurality of services to manage replication of data contained within the directory server from the supplier server to the consumer server, maintaining a change log of data replicated to the consumer server, updating data replicated to the consumer server, and resolving conflicts of the replicated data using a time stamp to determine the consumer server holding the most recent version of the replicated data.

[0017] In general, in one aspect, the invention involves an apparatus for replicating data in a directory server having a supplier and a consumer server. The apparatus comprises means for determining a need to replicate data in the directory server, means for using a plurality of services to manage replication of data contained within the directory server from the supplier server to the consumer server, means for maintaining a change log of data replicated to the consumer server, and means for updating data replicated to the consumer server.

[0018] Other aspects and advantages of the invention will be apparent from the following description and the appended claims.

Brief Description of Drawings

[0019] Figure 1 illustrates a block diagram of iPlanet™ Internet Service Development Platform.

[0020] Figure 2 illustrates part of a typical directory.

[0021] Figure 3 illustrates the LDAP protocol used for a simple request.

[0022] Figure 4 illustrates a directory entry showing attribute types and values.

[0023] Figure 5 illustrates a typical computer with components.

[0024] Figure 6 illustrates a default directory tree for iPlanet™ Directory Server in accordance with one or more embodiments of the present invention.

[0025] Figure 7 illustrates a block diagram of the replication architecture in accordance with one or more embodiments of the present invention.

Detailed Description

[0026] Specific embodiments of the invention will now be described in detail with reference to the accompanying figures. Like elements in the various figures are denoted by like reference numerals for consistency.

[0027] The invention described here may be implemented on virtually any type computer regardless of the traditional platform being used. For example, as shown in Figure 5, a typical computer (130) has a processor (132), memory (134), among others. The computer (130) has associated therewith input means such as a keyboard (136) and a mouse (138), although in an accessible environment these input means may take other forms. The computer (130) is also associated with an output device such as a display (140), which also may take a different form in a given accessible environment. The computer (130) is connected via a connection means (142) to a wide area network (144), such as the Internet.

[0028] A basic directory tree, also known as directory information tree (DIT), mirrors a tree model used by most file systems, with a tree root, or first entry, appearing at the top of a hierarchy. At installation, the iDS creates a default directory tree as show in Figure 6. The default directory tree contains a root (160) (dc=root, dc=suffix) and two entries. A first entry is o=NetscapeRoot (162). The data contained by this subtree is used by the iPlanet™ Administration Server. The iPlanet™ Administration Server handles authentication, and all actions that cannot be performed through LDAP (such as starting or stopping). A second entry is cn=config (164). This subtree contains iDS configuration information.

[0029] The initial directory tree contains one subtree reserved for the server itself and one subtree for iPlanet™ Administration Server. All the iDS typically contain the cn=config data, but only one (the first server installed) contains the o=NetscapeRoot information. The default directory can be built upon to add any data relevant to a directory installation.

[0030] Replication is the mechanism that automatically copies directory data from one directory server to another. Replicating a directory's contents increases the availability and performance of the directory and addresses the physical and geographical location of stored data. Using replication serves to copy any directory information tree or subtree (stored in a database) between servers. The directory server that holds the master copy of the information automatically copies any updates to all replicas. Replication enables the provision of a highly available directory service and the geographically distribution of data.

[0031] By replicating directory information trees to multiple servers, the directory is available even if some hardware, software, or network problem prevents directory client applications from accessing a particular directory server. Clients may also consult another replica. Note that to allow clients uninterrupted access to an updateable replica, a multi-master replication environment is needed, where

two or more servers hold a copy of the same read-write replica, and each server maintains a change log for the replica. Modifications made on one server are automatically replicated to the other server. In case of conflict, a time stamp is used to determine which server holds the most recent version. By replicating the directory tree across servers, the access load on any given machine may be reduced, thereby improving server response time. Replication makes it possible to own and manage a set of data locally while sharing the set of data with other directory servers.

[0032] A server that holds a replica that is copied to a replica on a different server is called a supplier server for that replica. A server that holds a replica that is copied from a different server is called a consumer server for that replica.

[0033] A replication architecture (RA) for a directory server is a group of services, functions, etc. that aid in the act of replication, *e.g.*, copying directory trees or subtrees from supplier servers to consumer servers. Figure 7 illustrates a diagram of the RA services within the RA structure. The RA (200) includes a Change Sequence Number (CSN) service (202), an Update Resolution Procedure (URP) Service (208), a Replica Update Vector (RUV) service (204), a Replication Agreement Service (210), a Server Initiated Replication Protocol (SIRP) Service (206), a Incremental Update Algorithm (IUA) Service (212), and a UniqueId service (214).

[0034] One service of the RA is the CSN Service. The CSN service creates, for each attribute value in each entry of the DIT, a unique combination of four numbers used to unambiguously determine the order in which update operations were received from clients. An URP service is then used to determine the correct ordering of various operations. The URP service uses the CSN service to determine the ordering of the operations by comparing the supplier server's CSN's

to the consumer server's CSN's. CSN's provide a global logical time, which gives a basis for ordering operations that were initially performed at different servers.

[0035] Another service is the RUV service. The RUV service indicates how up-to-date one replica is with respect to all other replicas. The RUV includes one CSN for each known replica that describes the latest update received from that replica. When one replica sends changes to another, the replica consults the consumer server's RUV and the RUV service determines the smallest set of updates that need to be sent to bring the replica up-to-date.

[0036] Another service is the Replication Agreement service. The Replication Agreement service is a DIT entry that describes the relationship between a supplier server and a single consumer server. Information present in the Replication Agreement service includes a portion of the DIT being replicated, a bind method and credentials used to authenticate to the consumer server, a replication schedule, etc.

[0037] A further service is the SIRP. The SIRP carries the state information for the URP service. The IUA service compares the supplier server and consumer server RUV's to properly order the update sequence from an iDS Change Log, which is a sequence of change records. The UniqueId service assigns a unique identifier to each entry added by the client. The UniqueId identifies the entry even if its distinguished name changes. The state information is information for an entry that includes the UniqueId and the set of CSNs for that entry.

[0038] While the prior discussion has involved incremental update protocols, one skilled in the art may appreciate that other methods exist, such as a total update protocol (TUP). Generally, the TUP is used to initialize or re-initialize a replica. The TUP carries all of the DIT entries along with the associated state information for each entry, and is sent as a series of LDAP extended operations.

[0039] The IUA reads the consumer's RUV, processes it through the changelog, and sends to the consumer any changes that the consumer has not yet received. All changes are sent even if the changes originated at some other replica. Because all changes are sent, transitive replication topologies are allowed. For example, suppose A & B are both updateable replicas. A replicates to B which replicates to consumer C. A does not directly replicate to C, but since B passes along changes originally received by A, C gets all changes originated at A and B.

[0040] Advantages of the present invention may include one or more of the following. An advantage of the RA services is that they provide additional functionality and control for each operation. Another advantage is that the RA services provide more efficient updates of the DIT. A further advantage of the RA is that it allows construction of replication environments that function even in the face of unavailability of an updateable replica. Another advantage of the RA is that it allows more flexible replication topologies. A further advantage of the RA is that in a multi-mastered environment, clients need only have connectivity to a single updateable replica. Those skilled in the art will appreciate that the present invention may have further advantages.

[0041] While the invention has been described with respect to a limited number of embodiments, those skilled in the art, having benefit of this disclosure, will appreciate that other embodiments can be devised which do not depart from the scope of the invention as disclosed herein. Accordingly, the scope of the invention should be limited only by the attached claims.